



# Smart Mandalorian

## **Integrantes:**

Santiago Duque Chacón  
2180099

Carlos Mauricio Tovar Parra  
1741699

## **Docente:**

Oscar Fernando Bedoya Leyva, PhD.

Universidad del Valle  
Facultad de Ingeniería  
Escuela de Ingeniería de Sistemas y Computación  
Inteligencia Artificial

# Capítulo 1

## Introducción

El problema planteado de forma específica para este proyecto presenta un ambiente correspondiente al *laberinto*, un agente que llamaremos *Mando*, una meta caracterizada como *Grogu*, una bonificación o mejora que es el elemento *Nave* y distintos obstáculos representados como *enemigos*. En conjunto, todos los elementos que forman el problema requieren de una visión particular para dar la solución a este, la cual es el camino que debería tomar el agente para llegar a la meta.

Se implementan 5 algoritmos de búsqueda que fueron abordados a lo largo del curso, los cuales se separan en las categorías de *No informados* e *Informados*. En la categoría de los algoritmos de búsqueda no informada encontramos 3 algoritmos:

- Preferente por Amplitud (*Breadth First Search*)
- Preferente por Profundidad (*Depth First Search*)
- Costo Uniforme (*Uniform Cost Search*)

Por otro lado, en la categoría de búsqueda informada se encuentran 2 algoritmos:

- Avara (*Greedy Search*)
- A\* (*AStar Search*)

Particularmente, en este informe se explicará la función heurística que se diseñó para implementar los 2 algoritmos de búsqueda informada. Esta función es la base de ambos algoritmos para hallar soluciones efectivas a la vez que toma en cuenta los elementos que constituyen el problema.

# Capítulo 2

## La heurística

### 2.1. Definición y explicación de la función

La función heurística implementada gira en torno a 2 factores principales, los cuales son Mando y la Nave. De esta manera, la función usa como base una heurística bastante común en este tipo de problemas, la cual consta de un simple cálculo de la distancia respecto a la meta; en nuestro caso optamos por usar la distancia de manhattan entre 2 puntos para este propósito en lugar de la distancia euclidiana, dada la condición de que el laberinto está conformado por celdas y posee una forma de cuadrícula.

Teniendo esto en cuenta, la función definida pasó por varias versiones en las que cada vez se refinaba un poco, en búsqueda de una heurística que sea admisible a la vez que no presentaba un optimismo demasiado elevado respecto a las soluciones reales, ya que esto podría implicar en una menor calidad de la solución. Por ejemplo, uno de los estados por los que pasó la función fue hacer un cálculo de  $h(g) = dist\_manhattan/2$ , lo cual suponía a Mando usando la Nave en la totalidad del recorrido, que da una heurística admisible, sin embargo como la Nave tiene una cantidad máxima de usos determinada por el combustible, esta heurística puede hacer una predicción demasiado optimista en algunos casos, como por ejemplo, al tener grandes distancias iniciales entre Mando y Grogu.

De esta forma y pasando por algunas otras versiones, la función final que se estableció es la siguiente:

$$\begin{aligned} m_{MG} &= \text{manhattan entre Mando y Grogu,} \\ m_{MN} &= \text{manhattan entre Mando y la Nave,} \\ m_{NG} &= \text{manhattan entre la Nave y Grogu,} \end{aligned}$$

$$h(n) = \begin{cases} \begin{cases} \min\{m_{MG}, m_{MN} + m_{NG} * 0,5\} & \text{si } m_{NG} \leq 10 \\ \min\{m_{MG}, m_{MN} + m_{NG} - 5\} & \text{sino} \end{cases} & \text{si Nave libre} \\ \begin{cases} m_{MG} * 0,5 & \text{si } m_{MG} \leq 10 \\ m_{MG} - 5 & \text{sino} \end{cases} & \text{si Mando en Nave} \\ m_{MG} & \text{sino} \end{cases} \quad (2.1)$$

La función se divide en 3 casos principales, los cuales están determinados por el estado en que se encuentre la Nave, siendo: la Nave está en el ambiente y Mando no la ha tomado (caso 1), Mando está en la Nave (caso 2), y por último, la Nave no está en el ambiente ni Mando está en la Nave (caso 3).

En el caso 1, se realiza otra verificación con el fin de ajustar un poco más el cálculo que se da en esta situación, revisando la distancia de manhattan desde la posición de la Nave hasta Grogue, donde hay 2 sub-casos:

- Si Grogue está a una distancia en la que la Nave puede llegar sin gastar la totalidad de su combustible ( $\leq 10$  unidades), la heurística devuelve el mínimo entre la distancia de manhattan Mando-Grogue y la distancia de manhattan Mando-Nave más la mitad de la distancia manhattan Nave-Grogue, lo cual corresponde a dirigirse a la Nave y el efecto de usarla.
- Si no se cumple la anterior condición, es porque Grogue está a una distancia mayor que el combustible máximo de la Nave ( $> 10$  unidades), luego la heurística devuelve el mínimo entre la distancia de manhattan Mando-Grogue y la distancia de manhattan Mando-Nave más la distancia manhattan Nave-Grogue menos 5, lo cual corresponde a dirigirse a la Nave y el efecto de usarla, con la diferencia de que aquí se contempla haber agotado el combustible antes de encontrar a Grogue.

Para el caso 2, la verificación que se realiza es similar a la del caso 1, revisando la distancia de manhattan desde la posición de Mando (que en este caso, está en la Nave) hasta Grogue. De esta manera se dan otros 2 sub-casos:

- Si Mando está a una distancia menor o igual a 10, se puede estimar que usará como máximo la totalidad del combustible, por lo que la heurística devuelve la mitad de la distancia de manhattan Mando-Grogue, de forma similar al sub-caso 1 del caso 1.
- Si no se cumple la anterior condición, es porque Grogue está a una distancia mayor que el combustible de la Nave, con lo que se puede estimar que el combustible se agotará, con lo que la heurística devolverá la distancia de manhattan Mando-Grogue menos 5, también de forma similar al sub-caso 2 del caso 1.

- Finalmente, el caso 3 se da cuando ninguna de las 2 condiciones principales se cumplen. Esto quiere decir que el caso 3 se presenta cuando: **a)** la Nave ya no está en el ambiente, y **b)** Mando no está en la Nave. Esto quiere decir, que el caso 3 se presenta únicamente cuando Mando ya ha usado la Nave y aún no ha llegado hasta la posición de Grog. En este caso la heurística devuelve simplemente el valor de la distancia de manhattan Mando-Grog.

## 2.2. Justificación de su admisibilidad

Previo a proceder con la justificación sobre el cumplimiento del criterio de admisibilidad para la función heurística planteada, es necesario recordar exactamente *qué entendemos* por función heurística. Para este propósito tomaremos la definición que brindan Norvig y Rusell [1], la cual consiste en comprender la función heurística como una estimación del mejor camino a la solución, sin embargo, no es más que eso, una estimación, por lo que en otras palabras, la heurística guía el algoritmo por el camino que *aparentemente* es el mejor. Así, la heurística es una función que por naturaleza es optimista, que dado un nodo de entrada y dependiendo únicamente del estado de ese nodo, estima un valor menor o igual hasta el nodo meta u objetivo.

Teniendo en cuenta este concepto, falta definir la admisibilidad de la heurística; para que una heurística  $h(n)$  sea admisible, debe cumplir con 2 condiciones: dado el nodo meta  $m$ , el cálculo de  $h(m)$  debe ser 0, y el valor de  $h(n)$  para cualquier otro nodo que no sea la meta, debe ser menor o igual al coste real de la solución obtenida.

$$\{(Si\ m \equiv Meta \Rightarrow h(m) = 0) \wedge (\forall n \in Ambiente : h(n) \leq costo\_solucion)\}$$

Entonces tomando estas definiciones, se puede considerar que la heurística planteada en el proyecto efectivamente es admisible ya que:

- Si el nodo a evaluar coincide con el nodo meta, se traduce a que Mando se encuentra en la posición de Grog. Esto quiere decir que al calcular la distancia de manhattan Mando-Grog será 0 e independientemente del estado del nodo para el resto del ambiente, en cualquiera de los 3 casos de la función heurística se retornaría 0.
- Para la segunda condición, se puede tratar de analizar cuál sería el arquetipo de un estado ideal en el problema: el mejor caso se daría siempre que exista un camino despejado (tanto de enemigos como muros) desde Mando hasta Grog siguiendo únicamente un sentido en cada eje o en su defecto, seguir un único sentido en un solo eje (el otro, ya sea  $x$  o  $y$  coincide con el de Grog), con lo que el valor de la distancia de manhattan será igual a la cantidad de movimientos que hará Mando. Dado este caso, el mayor valor que podría devolver la heurística es exactamente el valor de la distancia de manhattan Mando-Grog (según el estado para el resto del ambiente, podría incluso dar un valor menor según la situación de la Nave).

De esta forma, se puede evidenciar claramente que la heurística planteada es admisible y por tanto válida para su implementación en el proyecto.

## **2.3. Consideraciones finales y conclusiones**

La principal característica y ventaja de esta versión final para definir la función heurística, radica en considerar el efecto potencial que tiene la Nave para alcanzar la meta, pero manteniendo los valores en un rango suficiente para no producir una sobreestimación del valor a la vez que se logra cumplir con la condición de admisibilidad para la heurística.

Respecto a versiones anteriores y más simplificadas para la definición de la función, esta versión demuestra un comportamiento igual o con ligeras mejoras en ambos algoritmos informados para casos en los que el camino a recorrer es relativamente corto, sin embargo, a medida que crece la distancia real de la solución final, se logra observar el efecto positivo de las consideraciones tomadas, ya que progresivamente la cantidad de nodos expandidos se reduce frente a los que se expanden con otras definiciones de la función.

# Bibliografía

- [1] Peter Norvig y Stuart Russell. *Inteligencia Artificial: Un Enfoque Moderno*. 2nd. Madrid: Pearson, 2009. ISBN: 978-84-7829-137-3.